A RE-IMPLEMENTATION OF A BIOLOGICALLY-INSPIRED SLEEP ALGORITHM FOR INCREASED GENERALIZATION AND ADVERSARIAL ROBUSTNESS IN DEEP LEARNING NETWORKS

Laiyuan Zhang lz4y19@soton.ac.uk

Toby Coleman tsc1n19@soton.ac.uk

James Bygrave jb16g19@soton.ac.uk

ACKNOWLEDGEMENTS

We would like to thank Timothy Tadros for providing some of the MATLAB code and advice that helped re-implement aspects of the research described in this report.

ABSTRACT

Standard neural networks are vulnerable to adversarial attacks and distortions in their input. Both of these can detrimentally impact their performance and robustness. This report attempts to reproduce existing research that supports the belief that adding a sleeping phase to training can help deep neural networks avoid these mistakes. A series of different networks, including an original sleep algorithm, are analysed and their classification accuracy compared. This report shows a sleeping phase can certainly be effective at combating general distortions to the input, but struggles to prove its robustness against FSGM. The network that generalises best on noisy training data is the network trained on similar data. This report confidently supports some of the conclusions made by earlier authors, but highlights some areas of the underlying research that should be investigated further.

1 INTRODUCTION

In this report we re-examine and reproduce new recent research introduced at the International Conference on Learning Representations 2020 (ICLR 2020). The chosen paper questions whether adding a 'sleep' phase to an artificial neural network (ANN) can improve its performance and robustness to adversarial attacks and input distortions. This hypothesis is based on the author's belief that sleep is the reason why humans are so successful at classification tasks and that modelling the Spike Time Dependent Plasticity (STDP) of mammalian brains could help strengthen memory (Payne et al. (2009)) and increase the generalisation performance of deep learning models as well (Tadros et al. (2019)).

This report aims to fairly evaluate the methodology of the original research, attempt to reproduce the original research and highlight any discrepancies in the results and the reasons as to why they might exist. The core code of the algorithm that supports the findings documented in this report can be found at https://github.com/COMP6248-Reproducability-Challenge/COMP6248-Reproducability-Challenge-Sleeping-Algorithm. It is encouraged that the original paper (Tadros et al. (2019)) is read before this one in order to better understand the theory and background that underpins both papers.

1.1 LANGUAGE AND ENVIRONMENT

The re-implementation of the research, originally developed in MATLAB, was written with Pytorch 1.4.0 and Python 3.6 using the Torchbearer framework. It was ran using the Google Colab cloud service and executed on a Tesla K80 GPU.

1.2 EXPERIMENT SCOPE

A lack of processing capability, data storage and time means that the re-investigation within this report has chosen to ignore the DeepFool, JSMA and Boundary attacks. Experiments are not done on either the CUB200 or toy data set. In the case of the CUB-200 data set, its size and high resolution meant it was simply too big to store.

2 Implementation Details

2.1 Control Network

The first part of the research that was recreated was the control network. This network is important for constructing the other, more sophisticated, networks in addition to providing the baseline performance measure necessary to compare them. The parameters used in the architecture and training of the control network were kept the same with both hidden layers containing 1200 hidden units and using ReLU activation units. Each greyscale, 28x28, MNIST input image was flattened into a single vector of size 784 and fed as input to the network. The new control network followed the original paper and correctly utilises dropout and momentum to help prevent the model over-fitting the training data and help its stochastic gradient descent optimization converge faster while learning. On the other hand, these parameters were chosen by the previous author using a genetic algorithm with no documented design and no hope of being reproduced. As a result, we are forced to assume these values are close to optimal. It would have been nice, instead, to see how they were computed in the original paper.

2.2 Sleep Algorithm Network

Based on correspondence with the original author, we chose to modify some parameters to achieve a more desirable result within the new environment. A large number of experiments were performed to obtain a suitable set of parameters. The neuronal firing thresholds for each layer were changed from (36.18, 23.36, 36.68) to (15.579, 0.35, 16.52). The original threshold values were found to be too large and caused too few weights to satisfy the condition, so that the network was almost unchanged. It was important to stop an excessive number of parameters being updated or allowing the network to be unchanged. Additionally, the increase and decrease factors were changed to suppress any unwanted fluctuations of the weights. The increase factor was adjusted from 0.063 to 0.0065 and the decrease factor reduced by a factor of ten, to 0.0069.

A major challenge was the memory resources required during operation to traverse the network to find the weights that satisfied the update conditions. Although we kept the sleep duration as 27105 (meaning 27105 images were presented during sleep), we constrained the total number of weight updates to avoid memory overflow. Once this limit is reached, the algorithm is aborted.

Modifications were further made to the rules of the STDP function. Based on our experiments it was found that inverting this step led to better results. Weights were decreased if pre-synaptic spikes and post-synaptic spikes were both 1, and increased if their values were instead opposite. Using the original algorithm gave poorer results that were even worse than the control ANN. This decision was chosen based solely on the improvements to the empirical findings and with no real purposeful intuition. Clearly this change undermines the original paper and cannot be explained here. The answer for the questions this raises should be looked at in further detail.

2.3 Defensive Distillation Network

The defensive distillation network consists of connected teacher and student networks (Papernot et al. (2016)). The original research included a softmax function with added temperature term (T) used to the control the size of each probability and the influence of inputs on output layer activation. The value of T in the chosen paper was equal to 50. However, because attention was more focused on the sleep algorithm, this temperature hyper-parameter was neglected for convenience. Instead, results generated by the teacher network were passed straight to the student network. Later results can be used to argue that this decision had a negligible effect.

2.4 Distortion Networks

Two additional fine-tuned networks, using the pre-trained ANN, were trained on perturbed versions of the training data using blur and Gaussian noise. MNIST images were blurred with a 2D Gaussian kernel with a varying standard deviation between 0 and 2.5. Gaussian noise was generated with a variance value sampled between 0 and 1.0. One early concern was that the same noise variance led to less disturbed images compared with those from the original paper, likely because of the change in development environment. It was feared that this may lead to an overly positive result. This was not a problem for the blurred images.

2.5 Fast Gradient Sign Method (FGSM) Attack

FGSM Attack is a white-box offensive method that exploits knowledge of network architecture, training examples or gradient in order to force it into wrongly classifying data. These adversarial attacks are often invisible to the human eye and therefore pose a serious risk for standard ANNs. A shared, FGSM-attacked test set was used to validate each network.

3 Results and Analysis

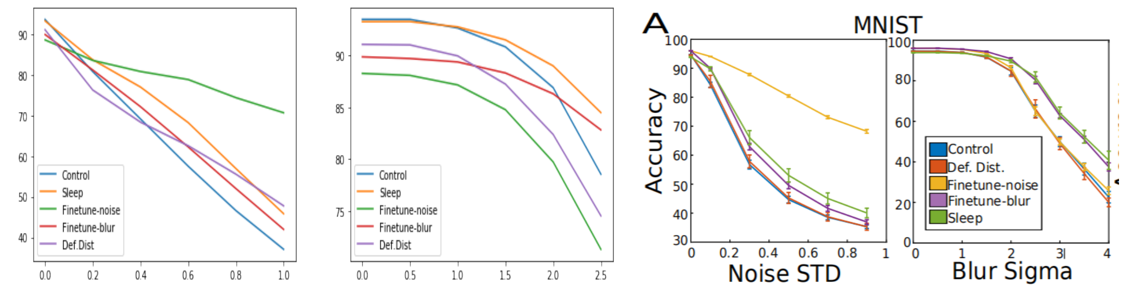
3.1 Total Score for an Attack (S_A)

This report decides to use the classification accuracy as the preferred performance metric since it is more easily comparable and clearer to interpret graphically.

3.2 Classification Accuracy

3.2.1 Generalised Distortions

The test set, for each network, was distorted with the same standard deviation or variance value for each specific disturbance before being increased. Our results are shown in figure 1.



a) Improved Noise Result b) Improved Blur Result c) Original Distortion Results

Figure 1: Improved classification accuracy of each network on distorted test sets.

Despite achieving a lower initial classification accuracy, our experiments were able to show that the fine-tuned noise network was the most robust to increasing amounts of added Gaussian noise. The plot of figure 1(a) bears a striking similarity to the original results shown in figure 1(c) with both obtaining roughly 70% classification accuracy for the same variance set at 0.9. There remains some discrepancies in the shape of these curves but we produce irrefutable evidence that a network trained on noisy data is the best performing network out of the five in this specific case.

Although the similarities are not as obvious, our experiment using varying degrees of blur on the test set also corroborates the research. The two most robust networks to this disruption are the sleep algorithm and the ANN fine-tuned on blurred inputs. The final accuracy for both of these models, with the standard deviation at 2.5, lies between 80-85% for both papers. Our own experiments were further able to to recreate the disparity in the accuracy between the two groups of networks when the standard deviation took this value. The accuracy in this instance for the control network, defensive distillation network and fine-tuned noise network were much lower - between 60% and 75%.

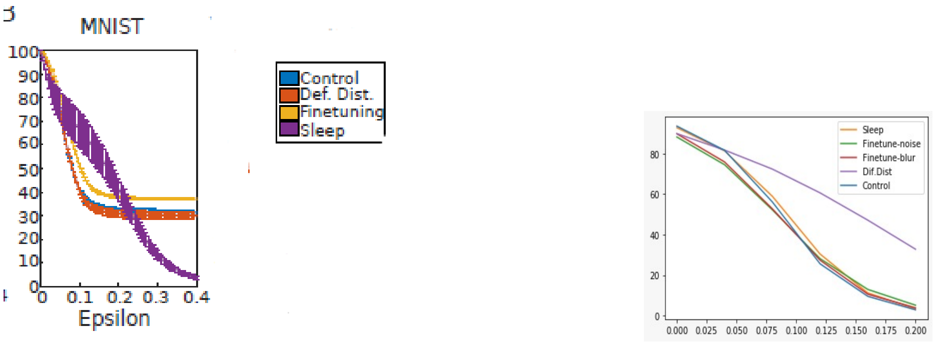
There was, however, a failure to assess the increase or decrease in correlation between closely related digits within the hidden layers of the network that utilised a sleeping phase. Consequently, we cannot reproduce all the results from the original research that analyse the effects of blur or noise on the sleep network.

Despite some small indications of subtle disagreement between the rate of degradation rate of each models' accuracy curves, figure 1 highlights the success of our research, but more importantly, goes a long way to verify the published paper. We are unable to contradict any of the conclusions made by the author based on our own results here.

3.2.2 FGSM Attack

The original claims made are that converting an ANN into a spiked neural network (SNN) with a built-in sleep phase can lessen the proportion of wrongly classified in the presence of a targeted FGSM attack. Evidence before this report shows that the amount of noise required to fool the sleep algorithm network was nearly double the amount that was used on the control and defensive distillation networks (Tadros et al. (2019)). This would indeed suggest greater robustness against certain adversarial attacks can be designed with a sleep period while learning.

Our own trials struggled to uphold this though. The recreated results we made differ drastically from the original results. The first major problem is that the defensive distillation network is shown to have the most resistance to FGSM attack in our results but is shown to be the weakest network in the findings from the ICLR 2020 paper. Neither does the re-implementation



a) Original results

b) Reproduced results

Figure 2: Classification accuracy of each network attacked with FGSM.

prove that the sleep algorithm network is twice as robust to FGSM when compared with the control network. Instead, all of the networks, except the distillation network, show similar classification performances as the value of ϵ is steadily increased.

Our results were gathered using smaller ϵ values taken between 0 and 0.2. Our developed networks simply refused to work for the same ϵ values used by the ICLR 2020 paper instead matching more closely to results from other academic papers (Goodfellow et al. (2014)). The Classification accuracy results when FGSM was configured with ϵ greater than 0.2 were catastrophic and did not reflect the previous research at all. We were forced to use a smaller range of values in order to compare our findings. One reason for this is the lack of a coherent or written description of how the FSGM attack was done from the original author. The bigger, and more likely error in our experiment, was the mistake of not removing images that the networks has already misclassified before applying the FGSM attack. There was a high chance this led to the poorer overall accuracy we obtained for each of the networks and was a significant oversight. Any future research should explicitly avoid repeating this mistake.

4 Conclusion

This report can confidently support the authors' hypothesis that a sleeping phase is capable of increasing network robustness for blur and noise disruptions. It also documents reproduced results that show each of these fine-tuned networks are only effective at combating the specific distortion they are trained on and not both kinds. Unfortunately, experiments were unable to reproduce the same robustness for any networks when faced with a FGSM attack. The report cannot confirm at this stage that a neural network embedded with the sleep algorithm is twice as resilient as a control or defensive distillation network in this scenario.

Future work should be focused on investigating some of the other adversarial attacks and defences included in the original paper that this report neglects. It was found that original experiments placed too high a cost on the resources we had available simply due to the size of the data as well as inefficiencies in the code that we struggled to optimise (e.g. nested loop statements). This report strongly recommends that these same experiments be tried again. It is hoped that we have shone a light on the potential of the research contained inside the ICLR paper and verify at least some of the conclusions it draws.

References

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

Jessica D Payne, Daniel L Schacter, Ruth E Propper, Li-Wen Huang, Erin J Wamsley, Matthew A Tucker, Matthew P Walker, and Robert Stickgold. The role of sleep in false memory formation. *Neurobiology of learning and memory*, 92(3):327–334, 2009.

Timothy Tadros, Giri Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Biologically inspired sleep algorithm for in-creased generalization and adversarial ro-bustness in deep neural networks. In *International Conference on Learning Representations*, 2019.